

1

DATA PACKET PROCESSING

2

FIELD OF INVENTION

3

The present invention is directed to a data packet
4 processing device for processing data packets and a method
5 thereto.

6

BACKGROUND:

7

One of the major challenges in processor design is the
8 optimization of the access latency to external memories.

9

Access latency is the time used by the processor to transmit
10 data via an interface to the external memory or receive data
11 from the external memory via the interface. Furthermore,
12 external memories often comprise DRAM memories which usually
13 are slow memory devices having a long access latency.

14 To speed up the access time for data packets, an internal
15 on-chip memory is usually provided to buffer parts of the
16 content of the slower external memory. This process is
17 called "caching" and the internal on-chip memory is called a
18 "cache".

19 The cache replacement strategies commonly used in general
20 purpose processors (GPP) are not appropriate for network
21 processors (NP) as the data access patterns of NP
22 applications differ significantly from GPP applications. In
23 general purpose processors the cache is provided with data

1 from addresses following the address actually being
2 processed.

3 The data access patterns for network processors differs
4 therefrom as the data packet are received from a network
5 continuously and the execution priority is determined after
6 reception of a data packet. Thus the memory addresses of the
7 data packets to be cached are independent from one another.
8 Furthermore the memory access has to be very flexible as the
9 data packet to be accessed next can change with every newly
10 received data packet.

11 The principle of speeding up data access by means of a
12 memory hierarchy has been integral to computers for a long
13 time. All major general purpose processors today use on-chip
14 caches. Possible cache replacement strategies include FIFO,
15 LRU and random.

16 In the documents US 5,651,002 and US 5,787,255, methods are
17 described to store packet headers in faster SRAMs while
18 storing the user data parts of packets the in DRAMs. Often,
19 there is no clear distinction between header and user data.
20 Thus, a system that speeds up access to only a small part of
21 the packet is not appropriate to speed up the processing of
22 the whole data packet.

23 **SUMMARY OF THE INVENTION:**

24 Therefore, it is an aspect of the present invention to
25 provide a smart processing strategy for a data packet
26 processing device, especially for a data packet processing
27 device to be located in a network. The above-mentioned

1 aspect is attained by the data packet processing device and
2 method for processing data packets described herein.

3 According to a first embodiment of the present invention, a
4 data packet processing device for processing data packets
5 received from a network is provided. The data packet
6 processing device includes a processor for processing said
7 data packets. An interface is operated to transmit data
8 packets to and from an external memory. A scheduler assigns
9 priority information to typically each of the received data
10 packets, wherein the priority information determines an
11 order of the data packets to be processed. The data packet
12 processing device further includes an internal memory to
13 store data packets. A memory manager is provided operable to
14 cause storing data packets in the external memory and to
15 provide data packets in the internal memory for processing
16 in the processor. Depending on the priority information of
17 the data packets, the memory manager provides the respective
18 data packets assigned by the respective priority information
19 in the internal memory for being processed by the processor.

20 The present invention has the advantage that data packets
21 are not preloaded in the internal memory in a manner known
22 from GPP which is not related to the priority of the
23 respective data packet. As it is determined by the
24 scheduler, the order of the processing of the data packets
25 can be used to store the data packets in the internal memory
26 to be processed next.

27 According to another embodiment of the present invention, a
28 method for processing data packets is provided. The data
29 packets which are received from the network are processed,
30 whereby a priority information is assigned to the received
31 data packets. The priority information determines an order

1 of the data packets to be processed. The received data
2 packets are stored in a fast accessible memory, wherein
3 depending on the priority information of the received data
4 packets, the respective data packets are provided in the
5 fast accessible memory for being processed or transferred
6 from a fast accessible memory to a main memory.

7 The method according to the present invention provides
8 optimized caching of data packets in a fast accessible
9 memory, e.g. a cache memory or another on-chip memory, which
10 is also called internal memory. While conventional methods
11 of caching data are directed to preload data at addresses
12 following to the actually executed address, the method
13 according to the present invention provides an option to use
14 the priority information determined by the scheduling means
15 not only for determining the order in which the data packets
16 are provided to the processing means but also to determine
17 the data packets that are to be available in the fast
18 accessible memory.

19 **BRIEF DESCRIPTION OF THE DRAWINGS:**

20 The foregoing and other aspects of these teachings are made
21 more evident in the following detailed description of the
22 invention, when read in conjunction with the attached
23 drawing figures, wherein:

24 Figure 1 shows a data packet processing device according to
25 an embodiment of the present invention; and

26 Figures 2a and 2b show flow charts of a method for
27 processing data packets according to another embodiment of
28 the present invention.

1 **DETAILED DESCRIPTION OF THE INVENTION**

2 This invention provides methods, apparatus and systems to
3 provide a smart processing strategy for a data packet
4 processing device, especially for a data packet processing
5 device to be located in a network. The invention includes a
6 data packet processing device and a method for processing
7 data packets described herein.

8 In an example embodiment of the present invention, a data
9 packet processing device for processing data packets
10 received from a network is provided. The data packet
11 processing device includes a processor for processing said
12 data packets. An interface is operated to transmit data
13 packets to and from an external memory. A scheduler assigns
14 priority information to typically each of the received data
15 packets, wherein the priority information determines an
16 order of the data packets to be processed. The data packet
17 processing device further includes an internal memory to
18 store data packets. A memory manager is provided operable to
19 cause storing data packets in the external memory and to
20 provide data packets in the internal memory for processing
21 in the processor. Depending on the priority information of
22 the data packets, the memory manager provides the respective
23 data packets assigned by the respective priority information
24 in the internal memory for being processed by the processor.

25 In conventional data packet processing devices, the
26 scheduling means is provided to determine the priority of
27 each of the received data packets to find out which data
28 packet should be processed next by the processing means.
29 Furthermore, in conventional general processing devices,

1 internal memories, e.g. caches are provided to speed up the
2 accessing of data by the processing means. Usually, an
3 internal memory is controlled by its own cache controllers
4 which decides by itself what data should be preloaded and
5 buffered in the internal memory.

6 The present invention now provides that information about
7 the priority of the received data packets given by the
8 scheduling means can be used by the controller for the
9 internal memory. The internal memory is preloaded with one
10 or more data packets from the external memory depending on
11 the priority information of the respective data packet, or
12 is preloaded with one or more data packets which are to be
13 transferred to the external memory for storage purposes, but
14 now are kept stored in the internal memory due to high
15 priority which means that they are to be processed next.

16 The present invention has an advantage that data packets are
17 not preloaded in the internal memory in a manner known from
18 GPP which is not related to the priority of the respective
19 data packet. As it is determined by the scheduler, the order
20 of the processing of the data packets can be used to store
21 the data packets in the internal memory to be processed
22 next.

23 Preferably, the memory manager loads a data packet stored in
24 the external memory into the internal memory depending on
25 the priority information of this data packet. This has the
26 advantage that the data packets having the highest priority
27 of all received data packets are transferred to the internal
28 memory to be processed as one of the next.

29 The memory manager can also transmit a received data packet
30 from the internal memory to the external memory depending on

1 the priority information of the data packet. As the received
2 data packets are usually stored in the internal memory, a
3 decision has to be made if the data packet should be kept
4 stored in the internal memory or be transferred to the
5 external memory to be stored in order to allow a quicker
6 data packet receipt. While the data packet is kept in the
7 internal memory if the priority is high and consequently it
8 is to be processed as one of the next, the data packet is
9 transferred to the external memory if the priority is low.

10 The internal memory has a size to store a number x of data
11 packets to be processed by the processing means, wherein the
12 priority of a data packet is high if the assigned priority
13 information indicates that the data packet is within the
14 next $x - 1$ ones to be processed. The priority of a data
15 packet is low if the assigned priority information indicates
16 that the data packet is not within the next $x - 1$ ones to be
17 processed. This allows an intensive use of the provided
18 internal memory, which allows an optimization of the access
19 of data packets.

20 According to another example embodiment of the present
21 invention, a method for processing data packets is provided.
22 The data packets which are received from the network are
23 processed, whereby a priority information is assigned to the
24 received data packets. The priority information determines
25 an order of the data packets to be processed. The received
26 data packets are stored in a fast accessible memory, wherein
27 depending on the priority information of the received data
28 packets, the respective data packets are provided in the
29 fast accessible memory for being processed or transferred
30 from a fast accessible memory to a main memory.

1 The method according to the present invention provides an
2 optimized caching of data packets in a fast accessible
3 memory, e.g. a cache memory or another on-chip memory, which
4 is also called internal memory. While conventional methods
5 of caching data are directed to preload data at addresses
6 following to the actually executed address, the method
7 according to the present invention provides an option to use
8 the priority information determined by the scheduling means
9 not only for determining the order in which the data packets
10 are provided to the processing means but also to determine
11 the data packets that are to be available in the fast
12 accessible memory.

13 To provide the respective data packets in the fast
14 accessible memory, the respective data packet has to be
15 transferred from the main memory - which can be a memory on
16 a separate chip, also called external memory - to the fast
17 accessible memory if the data packet is stored in the main
18 memory. If the data packet is already stored in the fast
19 accessible memory, the respective data packets should be
20 kept stored in the fast accessible memory and not be
21 transmitted to the main memory. Thus, it is possible that
22 data packets received recently are not transferred to the
23 main memory and then back to the fast accessible memory, but
24 are kept stored in the fast accessible memory, since their
25 priority is high and it is to be processed as one of the
26 next data packets.

27 Figure 1 shows a data packet processing device 1 according
28 to an embodiment of the present invention. The data packet
29 processing device 1 includes a processor 2 to process data
30 packets according to a given program code. The data packets
31 are received from a network 3 via a processor local bus 4. A
32 memory controller 5 is connected to the processor local bus

1 4 to receive the data packets from the network 3 and to
2 intermediately store the received data packet in an internal
3 memory 6 or in an external memory 7.

4 The internal memory 6 is a fast accessible memory, a
5 so-called cache memory. Storing the data packet in the
6 internal memory 6 allows a faster receipt of data packets
7 via a (not shown) data interface as the data packets can be
8 stored faster in the internal memory 6 than in the external
9 memory 7. The memory manager 5 is also connected to the
10 external memory 7, which is usually located outside of the
11 data packet processing device. The memory manager 5 and the
12 external memory are connected via an interface 10.

13 Data packets received from the network 3 are normally stored
14 in the fast accessible internal memory 6 and then
15 transferred to the external memory 7 controlled by the
16 memory manager 5. The received data packets are processed by
17 the processor 2, in an order determined by a scheduler 8.
18 Each of the received packets is examined upon receipt by the
19 scheduler 8 and a priority information is assigned to each
20 of the received data packets. The priority information
21 determines whether a data packet has a high or low priority.
22 The processor 2 is always provided by the data packets with
23 the highest priority of all received data packets and after
24 the respective data packet is completely processed, the data
25 packet with the next highest priority is provided to the
26 processor 2. Before the processor 2 can perform a function
27 to the respective data packets, the data packet has to be
28 loaded into the internal memory 6 from where parts of the
29 data packet or the whole data packet can be accessed faster
30 as it could be if the data packet was stored in the external
31 memory 7. According to the function of the internal memory
32 6 as a cache, it is desirable that while processing a data

1 packet at least the next data packet to be processed
2 according to the priority is loaded into the internal memory
3 6.

4 A scheduler 8 includes a pointer memory to store links
5 (pointers) to the data packet to be processed next. The
6 order of the pointers in the pointer memory is the order of
7 the respective data packets to be processed next. With the
8 receipt of each data packet from the network 3 the order of
9 pointers is actualized, so that after the processing of one
10 data packet the pointer with the address of the next data
11 packet to be processed is provided to the memory manager 5.

12 The transfer of data packets from the internal memory 6 to
13 the external memory 7 and from the external memory 7 to the
14 internal memory 6 is controlled by the memory manager 5.
15 This handling of the data packets is normally called caching
16 and is performed to give the processor 2 faster access to
17 the data packets when they are stored in the internal memory
18 6 as the access of data to the external memory 7 is slower.
19 The provision of the external memory 7 is necessary since
20 the number and the size of the received data packets
21 normally exceeds the capacity of the internal memory 6.

22 To speed up the data excess by the processor 2 the
23 respective data packet, which is actually processed and
24 preferably the data packet which is to be processed next
25 should be stored in the internal memory 6. The decision
26 which data packet should be loaded into the internal memory
27 6, is made by the memory manager 5 according to the
28 information in the pointer memory of the scheduler 8. The
29 processing order determined by the priority information
30 given by the scheduler 8 is provided to the memory manager 5
31 which then controls the preloading of the respective data

1 packets with the highest priority into the internal memory
2 6.

3 Basically, two options for the further proceedings are
4 usable, depending on where the respective data packet is
5 located. If the respective data packet is stored in the
6 external memory 7, the data packet is transferred to the
7 internal memory 6 controlled by the memory manager 5. If the
8 respective data packet was just received and stored in the
9 internal memory 6, the transfer of the respective data
10 packet to the external memory 7 is not performed. Instead
11 the respective data packet is kept stored in the internal
12 memory 6.

13 In some embodiments the internal memory 6 is divided up into
14 two sections. A first write section 61 is used to store
15 (buffer) the data packets just received from the network and
16 waiting for the memory manager 5 to transfer the respective
17 data packet from the write section of the internal memory 6
18 to the external memory 7. As the access to the write section
19 61 of the internal memory 6 is faster, normally received
20 data packets are first stored in the internal memory. But
21 storing a received data packet directly in the external
22 memory 7 is also possible. The second section, the read
23 section 62, is used to provide the data packets with the
24 highest priority to be processed, i.e. the data packet which
25 is actually processed by the processor 2 and the data packet
26 which are to be processed as the next data packets.

27 If a data packet is stored in the write section 61 and a
28 high priority is indicated by the pointer memory in the
29 scheduler 8, a transfer of the data package from the write
30 section 61 to the read section 62 can be performed. Also a
31 re-declaration of the write section 61 to a read section 62

1 could be useful. The write section 61 should be large enough
2 to also handle big data packets. Of course also a plurality
3 of write sections 61 can be provided. The read section 61 is
4 subdivided into one segment per pre-fetched data packet.
5 Normally, it should be sufficient to provide two read
6 segments. If more than one processor 2 is connected to the
7 processor local bus, two read segments 62 per processing
8 entity and two read segments to transmit data packets via
9 the network 3 should be sufficient. If the data packet
10 processing speed of the processor 2 is faster than the
11 preloading of data packets into the internal memory 6 it can
12 be advantageous to arrange more than two read segments 62
13 per processing entity on the internal memory 6.

14 In the first of the two read segments 62 the data packet
15 that is currently processed is stored and in the second
16 segment of the two segments the data packet that will be
17 processed next is stored. As soon as the processor 2
18 finishes working on one data packet and requests the next
19 one, the finished packet in the internal memory 6 is
20 replaced by the next packet in line after the now processed
21 packet. It may be possible that the processed data packet is
22 transferred via the network 3 or is stored in the write
23 section 61 or an additional provided write section 61 of the
24 interval memory 6 to be stored in the external memory 7.

25 The internal memory 6 has a size to provide enough memory
26 space for the several data packets to be stored. It is also
27 possible that if the size of the data packets is big, only
28 parts of the data packet are preloaded into the internal
29 memory 6. The bigger the capacity of the internal memory 6,
30 the bigger the parts of the data packets that can be
31 pre-fetched. It is therefore not necessary that complete
32 data packets have to be pre-fetched into the internal memory

1 6. It is also possible that the memory manager 5 first
2 fetches the head of each data packet. If either the
3 processor 2 has its own data cache or if the application
4 code works only once on each part of a packet, then the
5 memory manager 5 can purge any data from the internal memory
6 6 that has been read by the processor and replaces it with
7 another part of the data packet.

8 Figures 2a and 2b show flow charts illustrating a method of
9 the present invention. They show the handling of a data
10 packet received from a network 3 by the memory manager 5.
11 Referring to Figure 2a, when a data packet is received from
12 the network 3 (step S1), it is controlled by the memory
13 manager 5 directly stored in the internal memory 6,
14 preferably in the write section 61 of the internal memory 6
15 (step S2). While transferring the received data packet to
16 the internal memory 6, the scheduler 8 determines the
17 priority of the received data packet and provides priority
18 information assigned to the respective data packet. The
19 priority information of the received data packet and the
20 priority information of the stored data packets are taken
21 into account - if the priority information of the received
22 data packet is not self-explanatory - to determine an order
23 in which the data packets should be processed preferably.
24 The order of the respective data packet indicates if the
25 priority of the data packet is high or low (step S3).

26 If the priority of the data packet is high (step S4), the
27 received data packet is kept in the internal memory 6 to be
28 processed as one of the next data packets. In the next step
29 S5 the write section 61 of the internal memory 6 is
30 re-declared to a read section or the data packet is copied
31 from the write section 61 to the read section 62 to be
32 provided to the processor 2 to be processed as one of the

1 next data packets. If the write section is re-declared to a
2 read section 62, an available read section 62 has to be
3 defined as a write section so that the internal memory 6
4 provides enough buffer capacity for incoming data packets.
5 If the priority of the data packet is not high (step S4) in
6 the step S6 the data packet is transferred to the external
7 memory 7 by the memory manager 5.

8 In step S7, which follows steps S5 and S6, a check is made
9 to determine if a next packet is received which has to be
10 handled by the memory manager 5. If so, the procedure
11 returns to step S1. If no data packet is received it is
12 waited until the next data packet is received.

13 In addition the process according to Fig. 2b is processed.
14 The processor 2 is requesting the next data packet to be
15 processed. In step S8 it is detected if a read segment 62 of
16 the internal memory is available to be preloaded with a data
17 packet. This can be the case if the processor has fully
18 processed the actual data packet and transferred the
19 processed data packet to the network 3 or the write section
20 61 of the internal memory 6. If none of the read segments 62
21 is available, the process returns to step S8, otherwise it
22 proceeds with step S9. The data packet which should be
23 loaded in the available read section 62 is determined by the
24 pointer memory in the scheduler 8 in step S9. It is the data
25 packet with the next highest priority. As the respective
26 data packet is stored in the external memory 7 the data
27 packet is transferred to the internal memory 6, particularly
28 into the read section 62 which is ready to be loaded with an
29 new data packet (step S10). After step 10 the process
30 returns to step S8.

1 As the smallest possible configuration, one write section 61
2 and two read sections 62 are sufficient to perform the
3 method according to the present invention. While in one of
4 the read 62 sections the data packet which is currently
5 processed is stored in the other read section 62, the data
6 package which has to be processed next is stored. If the
7 processing of a data packet is faster than the preloading of
8 a data packet in the internal memory 6 it can be useful to
9 provide more than 2 read section 62 per processing entity
10 and network interface, respectively.

11 In some embodiments of the data packet processing device,
12 context information is assigned to each of the data packets
13 which has to be considered while processing the respective
14 data packet. In this case the internal memory should have a
15 size to store both the context information and the
16 respective data packet or a part of it to speed up the
17 access also to the context information.

18 In some embodiments more than two read sections 62 per
19 processor 2 are available, which are preloaded with data
20 packets which have to be processed as one of the next. The
21 decision if the priority of data packets is high is then
22 made as follows:

23 Given that the internal memory has a number x of read
24 sections 62 to store a number of data packets to be
25 processed. The priority of a respective data packet is
26 high if the assigned priority information indicates
27 that the data packet is within the next $x - 1$ ones to be
28 processed, i. e. besides the actually processed data
29 packet, which is stored in one of the read sections 62,
30 a number $x - 1$ of remaining read segments 62 is left to
31 store data packets with a high priority. The priority
32 of a data packet is low if the assigned priority

1 information indicates that the respective data packet
2 is not within the next $x - 1$ ones to be processed.

3 Variations described for the present invention can be
4 realized in any combination desirable for each particular
5 application. Thus particular limitations, and/or embodiment
6 enhancements described herein, which may have particular
7 advantages to the particular application need not be used
8 for all applications. Also, not all limitations need be
9 implemented in methods, systems and/or apparatus including
10 one or more concepts of the present invention.

11 The present invention can be realized in hardware, software,
12 or a combination of hardware and software. A visualization
13 tool according to the present invention can be realized in a
14 centralized fashion in one computer system, or in a
15 distributed fashion where different elements are spread
16 across several interconnected computer systems. Any kind of
17 computer system - or other apparatus adapted for carrying
18 out the methods and/or functions described herein - is
19 suitable. A typical combination of hardware and software
20 could be a general purpose computer system with a computer
21 program that, when being loaded and executed, controls the
22 computer system such that it carries out the methods
23 described herein. The present invention can also be
24 embedded in a computer program product, which comprises all
25 the features enabling the implementation of the methods
26 described herein, and which - when loaded in a computer
27 system - is able to carry out these methods.

28 Computer program means or computer program in the present
29 context include any expression, in any language, code or
30 notation, of a set of instructions intended to cause a
31 system having an information processing capability to

1 perform a particular function either directly or after
2 conversion to another language, code or notation, and/or
3 reproduction in a different material form.

4 Thus the invention includes an article of manufacture which
5 comprises a computer usable medium having computer readable
6 program code means embodied therein for causing a function
7 described above. The computer readable program code means
8 in the article of manufacture comprises computer readable
9 program code means for causing a computer to effect the
10 steps of a method of this invention. Similarly, the present
11 invention may be implemented as a computer program product
12 comprising a computer usable medium having computer readable
13 program code means embodied therein for causing a a function
14 described above. The computer readable program code means
15 in the computer program product comprising computer readable
16 program code means for causing a computer to effect one or
17 more functions of this invention. Furthermore, the present
18 invention may be implemented as a program storage device
19 readable by machine, tangibly embodying a program of
20 instructions executable by the machine to perform method
21 steps for causing one or more functions of this invention.

22 It is noted that the foregoing has outlined some of the more
23 pertinent objects and embodiments of the present invention.
24 This invention may be used for many applications. Thus,
25 although the description is made for particular arrangements
26 and methods, the intent and concept of the invention is
27 suitable and applicable to other arrangements and
28 applications. It will be clear to those skilled in the art
29 that modifications to the disclosed embodiments can be
30 effected without departing from the spirit and scope of the
31 invention. The described embodiments ought to be construed
32 to be merely illustrative of some of the more prominent

1 features and applications of the invention. Other
2 beneficial results can be realized by applying the disclosed
3 invention in a different manner or modifying the invention
4 in ways known to those familiar with the art.